Amendments to the Claims:

This listing of claims will replace all prior versions, and listings of claims in the application:

Listing of Claims:

1. (currently amended) A system for efficiently performing memory intensive computations, the system comprising:

a data cache located in memory of the system, wherein data stored in the data cache facilitates faster computations on the data stored in the data cache than if the data is stored in a database, wherein the data cache is coupled to a first set of data stored in a database and a second set of data stored in memory of the system, wherein the data cache is ~~and~~ configured to perform a scan operation on at least a portion of the first set of data and an update operation on the second set of data with changes that have occurred in the first set of data;

an engine manager coupled to the data cache and configured to instruct the data cache to perform the scan and update operations; and

a solver coupled to the data cache and configured to perform one or more computations on the updated second set of data stored in the data cache, the updated second set of data including the changes that have occurred in the first set of data.

2. (previously presented) The system of claim 1, further comprising an application specific plug-in coupled to the solver and configured to direct the solver to perform the computations on the second set of data.

3. (previously presented) The system of claim 1, wherein the system is configured to update the second set of data with substantially no more than the changes to the first set of data.

4. (previously presented) The system of claim 1, wherein the system is configured to update the second set of data with changes to the first set of data in a near-real-time fashion.

5. (previously presented) The system of claim 1, wherein the system is configured to update the second set of data with substantially no more than the changes to the first set of data that meet a given condition.

6. (previously presented) The system of claim 1, wherein the data cache coupled to a first set of data and a second set of data is coupled in a bidirectional fashion.

7. (previously presented) The system of claim 1, wherein the first set of data comprises metadata and application data.

8. (previously presented) The system of claim 1, wherein the solver comprises a generic algorithms module.

9. (previously presented) The system of claim 1, wherein the computations solve problems encountered in business applications.

10. (currently amended) A system for efficiently performing memory intensive computations, the system comprising:

a data cache having a second set of data, the data cache located in memory of the system, wherein data stored in the data cache facilitates faster computations on the data stored in the data cache than if the data is stored in a database, wherein the data cache is coupled to a first set of data stored in the database and, the data cache having a second set of data stored in memory of the system and configured to perform a scan operation on at least a portion of the first set of data and an update operation on the second set of data with changes that have occurred in the first set of data;

an engine manager coupled to the data cache and configured to instruct the data cache to perform the scan and update operations; and

a solver coupled to the data cache and configured to perform computations on the updated second set of data stored in the data cache, the updated second set of data including the changes that have occurred in the first set of data.

11. (previously presented) The system of claim 10, further comprising an application specific plug-in coupled to the solver and configured to direct the solver to perform the computations on the second set of data.

12. (previously presented) The system of claim 10, wherein the system is configured to update the second set of data with substantially no more than the changes to the first set of data.

13. (previously presented) The system of claim 10, wherein the system is configured to update the second set of data with the changes to the first set of data in a near-real-time fashion.

14. (previously presented) The system of claim 10, wherein the system is configured to update the second set of data with substantially no more than the changes to the first set of data that meet a given condition.

15. (previously presented) The system of claim 10, wherein the data cache coupled to a first set of data is coupled in a bidirectional fashion.

16. (previously presented) The system of claim 10, wherein the first set of data comprises metadata and application data.

17. (previously presented) The system of claim 10, wherein the solver comprises a generic algorithms module.

18. (previously presented) The system of claim 10, wherein the computations solve problems encountered in business applications.

19. (currently amended) A system for efficiently performing memory intensive computations, the system comprising:

a database comprising a first set of data;

a plug-in configured to provide application specific functionality; and

an in-memory engine located in memory of the system, wherein data stored in the in-memory engine facilitates faster computations on the data stored in the in-memory engine than if the data is stored in a database, wherein the in-memory engine is coupled to the database via a synchronization mechanism and comprising a second set of data stored in memory of the system, the in-memory engine configured to interface with the plug-in, the in-memory engine configured to perform computations on the second set of data to derive a first result, the in-memory engine configured to transfer, when the first set of data changes, these changes to the second set of data via the synchronization mechanism in order to update the second set of data and perform computations ~~thereon~~ on the updated second set of data to derive a second result.

20. (previously presented) The system of claim 19, wherein the system is configured such that the second set of data is updated with substantially no more than the changes to the first set of data.

21. (previously presented) The system of claim 19, wherein the system is configured to update the second set of data with the changes to the first set of data in a near-real-time fashion.

22. (previously presented) The system of claim 19, wherein the system is configured to update the second set of data with substantially no more than the changes to the first set of data that meet a given condition.

23. (previously presented) The system of claim 19, wherein the in-memory engine is coupled to the database in a bidirectional fashion.

24. (previously presented) The system of claim 19, wherein the database comprises metadata and application data.

25. (previously presented) The system of claim 19, wherein the in-memory engine comprises a data cache containing the second set of data.

26. (previously presented) The system of claim 19, wherein the in-memory engine comprises a generic algorithms module and an in-memory engine manager.

27. (previously presented) The system of claim 19, wherein the computations solve problems encountered in business applications.

28. (currently amended) A method of efficiently performing memory intensive computations, the method comprising:

performing, using a data cache, a scan operation on at least a portion of a first set of data;

performing, using the data cache, an update operation on a second set of data with changes that have occurred in the first set of data, the data cache located in memory, wherein data stored in the data cache facilitates faster computations on the data stored in the data cache than if the data is stored in a database; and

performing computations, using a solver, on the updated second set of data stored in the data cache, the updated second set of data including the changes that have occurred in the first set of data.

29. (previously presented) The method of claim 28, wherein the second set of data is created from substantially all of the first set of data.

30. (previously presented) (previously presented) The method of claim 28, wherein the second set of data is created from all of the first set of data.

31. (previously presented) The method of claim 28, wherein the second set of data is updated with substantially no more than the changes to the first set of data.

32. (previously presented) The method of claim 28, wherein the second set of data is updated with the changes to the first set of data in a near-real-time fashion.

33. (previously presented) The method of claim 28, wherein the second set of data is updated with substantially no more than the changes to the first set of data that meet a given condition.

34. (previously presented) The method of claim 28, further comprising updating the first set of data with the second set of data.

35. (previously presented) The method of claim 28, wherein the computations solve problems encountered in business applications.

36. (currently amended) A system for efficiently performing memory intensive computations, the system comprising:

scanning means for performing, using a data cache, a scan operation on at least a portion of a first set of data;

update means for performing, using the data cache, an update operation on a second set of data with changes that have occurred in the first set of data, the data cache located in memory, wherein data stored in the data cache facilitates faster computations on the data stored in the data cache than if the data is stored in a database; and

solving means for performing computations, using a solver, on the updated second set of data stored in the data cache, the updated second set of data including the changes that have occurred in the first set of data.

37. (currently amended) A computer-readable medium for efficiently performing memory intensive computations, the computer-readable medium comprising:

instructions for performing, using a data cache, a scan operation on at least a portion of a first set of data;

instructions for performing, using the data cache, an update operation on a second set of data with changes that have occurred in the first set of data, the data cache located in memory, wherein data stored in the data cache facilitates faster computations on the data stored in the data cache than if the data is stored in a database; and

instructions for performing computations, using a solver, on the second set of data stored in the data cache, the updated second set of data including the changes that have occurred in the first set of data.

38. (new) A system for efficiently performing memory intensive computations, the system comprising:

a data cache located in memory of the system, wherein data stored in the data cache facilitates faster computations on the data stored in the data cache than if the data is stored in a database, wherein the data cache is coupled to a first set of data stored in a database and a second set of data stored in memory of the system, wherein the data cache is configured to perform a scan operation on at least a portion of the first set of data and a first update operation on the second set of data with changes that have occurred in the first set of data;

an engine manager coupled to the data cache and configured to instruct the data cache to perform the first update operation; and

a solver coupled to the data cache and configured to perform one or more computations on the updated second set of data stored in the data cache, the updated second set of data including the changes that have occurred in the first set of data,

wherein the engine manager is configured to determine if the first set of data has changed since the last update operation, wherein if the first set of data has changed, the engine manager is configured to perform a second update on the second set of data in the data cache with the changes to the first set of data since the last update operation,

wherein the solver is configured to re-perform the one or more computations on the updated second set of data stored in the data cache including the changes that have occurred in the first set of data since the last update operation.